What is claimed is:

1. A process for simulating an input sequence comprising:

partitioning the input sequence into a partition including a set of substrings and a tail, wherein the substrings have lengths that are not all equal; and

outputting the substrings in a random order to generate an output sequence.

2. The process of claim 1, wherein the tail is selected from the group consisting of an empty string and the substrings of the partition.

3. The process of claim 1, wherein partitioning the sequence comprises selecting each of the substrings to consist of one or more consecutive symbols from the input sequence, where each of the substrings differs from the other substrings of the partition.

4. The process of claim 1, wherein for each substring, the substring is a shortest sub-sequence of consecutive symbols from the input sequence such that the substring differs from all of the substrings that are in the partition and preceding in the input sequence.

5. The process of claim 4, further comprising:

drawing a random integer from a range of $|T_X|$ integers, where $|T_X|$ is the number of sequences in a set T_X such that for each sequence in the set T_X, a partition of the sequence into substrings such that each substring is a shortest sub-sequence of symbols from the sequence that differs from all of the substrings of the partition that are preceding in the sequence includes a set of substrings that is equal to the set of the substrings in the partition of the input sequence; and

mapping the random integer to a corresponding one of the sequences in the set T_X, wherein the sequence corresponding to the random integer defines the random order for outputting the substrings.

6. The process of claim 1, wherein outputting the substrings comprises:

organizing the substrings in a tree having multiple levels, wherein each of the levels contains substrings of equal length, and branches between any two of the levels connect each substring in a higher of the two levels to a substring that results from deleting a last symbol of

the substring;

designating the substrings in the partition as available;

selecting one of the substrings as a current substring;

randomly selecting one of the branches from the current substring to the substrings in a higher one of the levels of the tree, wherein each of the branches from the current substring has a probability of being taken that depends on how many available uses there are of the substrings that are connected through the branch to the current substring;

changing the current substring to the substring at an end of the branch selected;

in response to the current substring not being available, repeating selection of one of the branches from the current substring and changing the current substring to the substring at the end of the branch selected; otherwise

outputting the current substring; and

marking the current substring as used.

7. The process of claim 6, wherein selecting one of the substrings as the current substring comprises selecting an empty string as the current substring.

8. The process of claim 6, wherein marking the current substring as used changes the string from being available to being unavailable.

9. The process of claim 6, wherein marking the current substring as used reduces available uses of the current substring.

10. The process of claim 6, wherein the probability of each of the branches being taken is equal to a ratio of a total of the available uses of the substrings that are connected through the branch to the current substring and a total of available uses of the substrings that are connected through all of the branches connecting the current substring to higher levels in the tree.

11. The process of claim 1, further comprising:

generating the input sequence from an ordering of pixel values in a digital representation of a texture; and

generating a digital representation of a simulation of the texture from the output

sequence.

12. The process of claim 1, further comprising:

generating the input sequence from measurements of a first system; and

using the output sequence for testing of a second system.

13. The process of claim 1, wherein outputting the substrings in a random order to generate the output sequence is performed in a computer.

14. A medium storing a computer program that implements a process comprising:

partitioning an input sequence into a partition including a set of substrings and a tail, wherein the substrings have lengths that are not all equal; and

outputting the substrings in a random order to generate an output sequence.

15. The medium of claim 14, wherein each of the substrings is a shortest sub-sequence of consecutive symbols from the input sequence such that the substring differs from all of the substrings that are in the partition and preceding in the input sequence, and the process implemented by the computer program further comprises:

drawing a random integer from a range of $|T_X|$ integers, where $|T_X|$ is the number of sequences in a set T_X such that for each sequence in the set T_X, a partition of the sequence into substrings such that each substring is a shortest sub-sequence of symbols from the sequence that differs from all of the substrings of the partition that are preceding in the sequence includes a set of substrings that is equal to the set of the substrings in the partition of the input sequence; and

mapping the random integer to a corresponding one of the sequences in the set T_X, wherein the sequence corresponding to the random integer defines the random order for outputting the substrings.

16. The medium of claim 14, wherein outputting the substrings comprises:

organizing the substrings in a tree having multiple levels, wherein each of the levels contains substrings of equal length, and branches between any two of the levels connect each substring in a higher of the two levels to a substring that results from deleting a last symbol of the substring;

designating the substrings in the partition as available;

selecting one of the substrings as a current substring;

randomly selecting one of the branches from the current substring to the substrings in a higher one of the levels of the tree, wherein each of the branches from the current substring has a probability of being taken that depends on how many available uses there are of the substrings that are connected through the branch to the current substring;

changing the current substring to the substring at an end of the branch selected;

in response to the current substring not being available, repeating selection of one of the branches from the current substring and changing the current substring to the substring at the end of the branch selected; otherwise

outputting the current substring; and

marking the current substring as used.

17. A process for generating a simulated sequence, comprising:

creating a tree structure having nodes that correspond to substrings resulting from parsing an input sequence, wherein all of the nodes except a root node are initially designated as being unused;

setting a current node equal to the root node;

ending the process if none of the nodes of the tree structure are unused;

in response to the current node being unused, outputting a substring corresponding to the current node as part of the simulated sequence, designating the current node as being used, and setting the current node equal to the root node; and

in response to current node being used, selecting a branch from the current node to one of the nodes in a higher level of the tree structure and setting the current node to the node at an upper end of the selected branch.

18. The process of claim 17, wherein the substrings resulting from parsing the input sequence comprises the substrings from parsing the input sequence according to the Lempel-Ziv incremental parsing rule.

19. The process of claim 17, wherein the input sequence comprises a binary sequence, and selecting the branch from the current node comprises:

selecting a first branch from the current node if a second branch from the current node

is blocked; and

selecting the second branch from the current node if the first branch from the current node is blocked.

20. The process of claim 17, wherein the input sequence comprises a binary sequence, and selecting the branch from the current node comprises selecting a branch Vb, wherein branch index b is a randomly drawn bit with a probability of being 1 equal to U(V1)/[U(V0)+U(V1)], U(V1) is a number of unused nodes on a branch V1 from the current node, and U(V0) is a number of unused nodes on a branch V0 from the current node.